Nokia 9110 Communicator
Setting Up Dial-In Service

TABLE OF CONTENTS

Nokia is a registered trademark of Nokia Corporation. Nokia’s product names are either trademarks or registered trademarks of Nokia. Other product and company names mentioned herein are trademarks or trade names of their respective owners.

Nokia Mobile Phones operates a policy of continuous development. Therefore we reserve the right to make changes and improvements without prior notice.

1. OVERVIEW

This document explains the basic requirements for supplying Internet or Intranet connection for the Nokia 9110 Communicator, and can be used by Internet Service Providers as well as companies planning Intranet access for the Nokia 9110 Communicator. This document does not cover all the possibilities for providing dial-in services for the communicator, but it can be used as a guide when setting up these services. All trademarks used in this document are the property of their respective owners.

2. ABOUT THE CONNECTION

Normal data transfer speed of the GSM network is 9600 bit/s. The communicator has a built in support for 14400 bit/s , and it can be used if it is supported by the GSM network.

The data transfer speed can be increased by using compression. The compression protocols supported by the communicator are discussed in chapter 2.3.

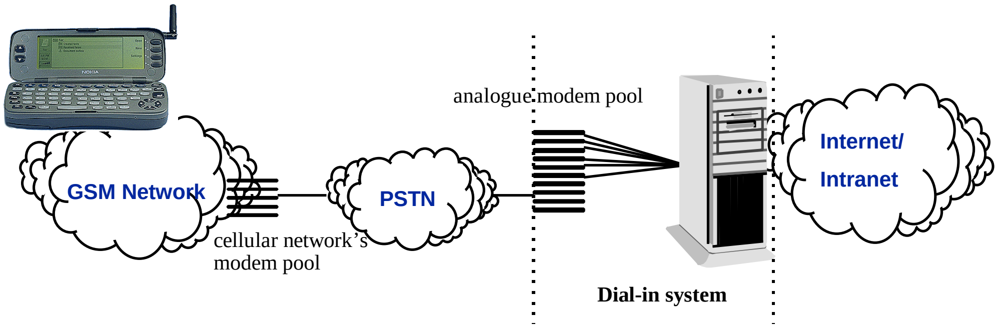
2.1. Requirements

A dial-in server for the Nokia 9110 Communicator should have a dial-in telephone line(s) with a Point to Point Protocol (PPP) service. There are also digital solutions available. Different methods for providing dial-in communicator service are explained in the following chapters.

By default the communicator uses a dedicated Point to Point Protocol (PPP) service. The dedicated PPP service means that the PPP state machine should be ready to receive PPP data before the username and password are given to the system. The log-in sequence can be customised using scripts when needed. More information about the scripts can be found in section 3 of this text.

The analogue modem access to Internet or Intranet with the communicator is shown in figure 2.1, below.

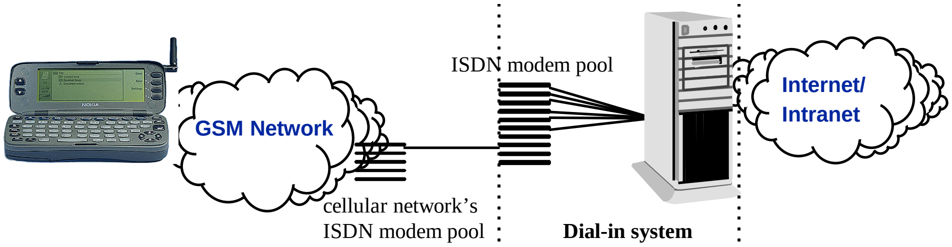
Figure 2.1, Analogue modem access with the communicator.



The analogue access requires data support from the cellular network. The dial-in system depends on the needs of the organisation. It can be as simple a modem connected to a Microsoft RAS server. Of course in the bigger organisations the dial-in system can be several modems attached to a PPP server with a separate server for user authentication. The connection time is typically up to 40 seconds.

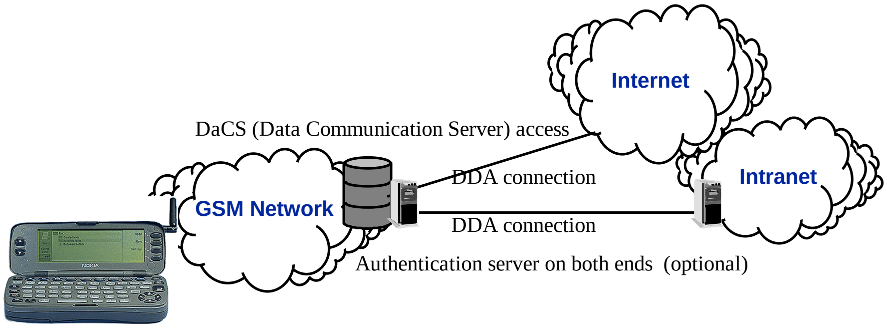
In figure 2.2, below, there is a description of digital ISDN connection. An ISDN connection provides shorter connection times (typically up to 15 seconds) and more reliable connections. This requires V110 or V120 protocol support from the GSM network operator and from the ISDN modem system. If the cellular operator does not have an ISDN modem pool set up for wireless data users, the ISDN connection will not work.

Figure 2.2, ISDN modem access with the communicator.



DaCS (Data Communication Server) in figure 2.3, below, enables the direct DDA (Direct Digital Access) connection from the GSM network to the Internet or to an Intranet. DaCS also provides a possibility to use ISDN connection instead of DDA. Nokia Telecommunications has a family of products, called the Nokia Artus family and DaCS is one part of the family. More information about the Artus family can be found at *http://www.nokia.com*.

Figure 2.3, Direct digital access with the communicator.


Internet
DaCS (Data Communication Server) access
DDA connection
Intranet
GSM Network
DDA connection
Authentication server on both ends (optional)

The DaCS provides a fully digital and non-modem DDA connection to the Internet or Intranet. This makes the connection reliable with short connection times (typically up to 15 seconds). The DaCS is ideal solution to big Internet Service Providers and large companies. Also operators have a possibility to offer services to their customers using the DaCS.

2.2. Configuring the TCP/IP

The TCP/IP configurations can be made manually to the communicator's Internet access point settings. Automating the configuration can make this set-up more user-friendly.

The communicator uses IPCP (rfc1332), DHCP (rfc1541), and /or BOOTP (rfc951) for IP-address negotiations. The dial-in system should provide Microsoft extensions for IPCP DNS negotiation (MSIPCP), BOOTP-server, or DHCP-server for automatically configuring the communicator's other TCP/IP settings, such as the DNS server.

The TCP/IP settings can also be sent to the communicator using a special short message containing Internet access information, such as username, password, phone number information and scripts. The format of these special short messages is presented in Using the Special SMS Features at *http://www.forum.nokia.com*.

2.3. Compression

The communicator supports V.42bis, but requires support from the GSM network and the modem called. This support is not yet commonly provided. If this support is provided, configure the communicator end for V.42bis through Internet Settings => Modem Initialisation section => "Autobauding with V.42bis compression". In the call-in-modem end the V.42bis can be set with the at command: "at+ds=3,0".

The communicator also supports PPP Stac LZS Compression Protocol (and its following modes: LCB, CRC, Sequence Number and Extended Mode) (RFC1974) and Microsoft Point-to-Point Compression (MPPC). To use these compressions the dial-in system must be configured properly for the usage.

2.4. Secure Authentication and Connections

Secure connections and password protection help to prevent unauthorised access to sensitive information. The communicator support's some methods which will secure the authentication. In the PPP authentication the communicator supports PAP and CHAP.

Third party authentication is one method for protecting passwords, for example, receiving a “one time” password (e.g., TACACS and RADIUS based systems). In the communicator end, the users only needs to set the password to be prompted at the Internet Access Point.

There are also some secure connection methods available. One of the most popular is SSL (Secure Socket Layer). Some banks have established on-line banking over the Internet (i.e. they use an http connection). SSL can also be used with Lotus Domino. SSL 3.0 software is available for the communicator. See the communicator support section of Forum Nokia for more information (*http://www.forum.nokia.com*).

3. USING SCRIPTS

This chapter specifies how to define custom login sequences for the time frame between the data call establishment (i.e. CONNECT response from the AT command interpreter) and the starting of the PPP protocol negotiations in the communicator.

3.1. Handling Scripts in the communicator

In the Internet application, press **Settings** → choose *Internet access* → press **New** or **Edit** → choose *Advanced settings* → choose *Login customisation* and enter your script as outlined in this document.

To add a script go to the *Login customisation* pop-up menu, press **New**, type the script and then press **Close**. Scripts can also be added using special SMS messages. For SMS message formats, please see the The remote configuration options at *http://www.forum.nokia.com*.

To choose an existing script, select a script, the *Login Customisation* pop-up menu and press **OK**. The same script can be chosen to be used with many Internet access points.

To Modify an existing script select a script in the *Login Customisation* pop-up menu, press **Open**, modify the script and then press **Close**.

To Delete an existing script select a script in the *Login Customisation* pop-up menu, press **Open**, press **Delete** in the editor and then press **Delete** in the confirmation dialog box.

3.2. Script Format

The communicator script language is a simple chat script language. It expects that the dial-in server sends a word to the communicator and the response which the communicator should send is separated with a space. For example:

```
Password: MyPassword
```

where the "Password:" is what the server sends and "MyPassword" is what the user is supposed to input.

Unless the string which the communicator sends ends with /c, a carriage return character (ASCII 0x0d) is sent after the string. If words contain spaces, they must be surrounded by "" quotation marks, e.g. "This word contains spaces". There must not be any characters outside of the quotation marks.

Chat scripts are:

```
expect send
expect send...
```

where the communicator expects something from the dial-in server and after the server has sent something the communicator sends something back.

```
or
expect1-send1-expect2 send2...
```

where if the "expect1" does not come from the server in TOUT (time out) time, "send1" is sent to the server. Then the script waits for the "expect2" from the server, when that is received "send2" is sent. If "expect1" does come then "send2" is sent. In the example scripts, case 2 and 3, these type of script conversations are used.

3.3. Script Features

Comments begin with the number sign (#), extend to the end of the line, and are ignored during script execution. Scripts are case-sensitive and do not have a specified maximum length.

Because of some possible problems there are two words which define the behaviour of the script in problem situations. ABT, the abort-string expects certain words coming from the server which would lead to abort the connection. Such strings may be: NO CARRIER or BUSY. The ABT word has to be placed to "expect" part of the script. If abort-string is seen in the incoming stream while executing the remainder of the chat script, the dialling attempt is aborted and user receives an error note.

The other word is TOUT, timeout-time. While executing the current chat script, the communicator waits timeout-time seconds for an expected response, before regarding the dialling attempt as failed. If no timeout-time is given in the script, the default read timeout of 45 seconds is used.

There are two embedded programs which can be used in the scripts, "ask" and "askq". The percent character (**%**), surrounds the name of an embedded program that is to be run before proceeding.

Program	Description
ask(<text>)	Shows user a text input dialog with title given in the paranthesis. Entered characters are echoed normally. When the **OK** button is pressed, dialog is dismissed and the entered text is sent onwards.
askq(<text>)	Same as **ask()**, but entered characters are echoed with asterisk characters ("*").

When the program has run the input will be sent to the dial-in server. "Ask" can be used when requesting for a username and "askq" when requesting for a password.

Time spent waiting for user input in **ask()** and **askq()** does not count towards “expect” or “send” time-outs. If the user doesn't dismiss the input dialog set up by **ask()** or **askq()** during the user timeout of 4 minutes, the operation is continued as if the user has dismissed the dialog.

There are several special character sequences which can be used with scripts. The table below will describe them.

Sequence	Description
""	Expect or send nothing. The sequence will be followed by a carriage return.
/s	Send or receive a space character (ASCII 0x20)
/t	Send or receive a horizontal tab character (ASCII 0x09)
/n	Send or receive a line feed character (ASCII 0x0a)
/r	Send or receive a carriage return character (ASCII 0x0d)
//	Send or receive a slash character (ASCII 0x5c)
/^	Send or receive a carat character (ASCII 0x5e)
^<character>	Send or receive the single character Ctrl-<character> (ASCII 0x00 through 0x1f)
/ddd	Send or receive a character specified in octal digits
/p	Pause for 0.25 seconds before proceeding (send only)
/d	Delay for 2.00 seconds before proceeding (send only)
/c	Don't append a carriage return character after sending the preceding string (send only)
/q	Don't print following send strings (e.g. a password) in any debugging or logging output. Subsequent /q sequences toggle `quiet' mode.
/A	Parse the incoming string as an IP address, written as four decimal numbers separated by periods, and use it for the local end of the point-to-point connection (receive only).
/D	Parse the incoming string as an IP address, written as four decimal numbers separated by periods, and use it for the primary name resolver during this connection (receive only).

3.4. How to Create a Script

If you do not know what kind of responses are requested by the dial-in server, there are few ways to find that out. One can try out to make a Terminal call to the dial-in number and see what the server requests.

The other way is to go to the communicator Internet access settings where you can select a script (as described in Handling scripts in the communicator). At the Login customisation screen, instead of picking a script you can select "Manual". This option will bring up a terminal screen in the login sequence.

When you have found what the dial-in server requests then you can start building a script. First it is important to add the ABT, abort strings, in case of problems. Then the timeout value should be set. After this it is time for the server responses and what the communicator should reply. Let's take an example. This is what the server requests:

Sender	Message
IAP	Welcome!<CR>
User	<CR>
IAP	Domain:
User	MyDomain
IAP	Login:
User	Myusername<CR>
IAP	Password:
User	2secret4u<CR>
IAP	Server-prompt>
User	PPP
IAP	(begins PPP)

First the user has to send a carriage return to the server, to start the whole sequence. Then a domain has to be set and after that a username and a password. First the abort and time out values.

```
ABT "NO CARRIER" ABT BUSY
TOUT 25
```

Then the communicator will start the script and send a carriage return, with out expecting anything from the server.

```
"" /r
```

After that the communicator waits for "Domain:" word coming from the server and when it has received it the selection "MyDomain" is sent to the server.

```
Domain: MyDomain
```

Then the communicator will ask for the user to insert username (Username: will appear on the user screen.) after the string "name:" comes from the server. Note that expected string can be a part of a longer string which comes from the server.

```
name: %ask(Username:)%
```

And the same thing with the password:

```
word: %askq(Password:)%
```

Then the server should give the prompt to the user, where the user should type in "PPP" in order to start PPP session.

```
> PPP
```

The whole script will look like this after it has been inserted to the communicator:

```
ABT "NO CARRIER" ABT BUSY
TOUT 25
"" /r
Domain: MyDomain
name: %ask(Username:)%
word: %askq(Password:)%
> PPP
```

In the following chapter there are more examples on the scripts.

3.5. Example Scripts

Note: <CR> denotes a carriage return is entered.

Case 1:

The user must execute the following dialog with the IAP to start PPP negotiations:

Sender	Message
IAP	Welcome to our Great Access service, Please enter your login name and password.<CR>
IAP	Login:

User	Myusername<CR>
IAP	Password:
User	2secret4u<CR>
IAP	(begins PPP)

```
# Great Access service login script
# In case the line is busy
ABT "NO CARRIER" ABT BUSY
# Timeout should be set if there are problems
TOUT 25
# If the server sends a string with ':' the communicator will send the username
":" Myusername
# If the server sends a string with ':' the communicator will send the password
":" 2secret4u
# End of script, PPP starts
```

Case 2:

The User must execute the following dialog with the IAP to start PPP negotiations:

Sender	Message
IAP	Welcome to our Great Access service, Please enter your login name and password.<CR>
IAP	Login:
User	Myusername<CR>
IAP	Password:
User	2secret4u<CR>
IAP	>
User	pppd -f xyz.opt<CR>
IAP	(begins PPP)

The following script will establish the connection:

```
# Great Access service login script
# In case the line is busy
ABT "NO CARRIER" ABT BUSY
# Timeout should be set if there are problems
TOUT 25
# Send CR if 'Login:' does not appear, if it does send the
# username 'Myusername'.
ogin:-""-ogin: Myusername
# Ask password from the user when the IAP sends a string
# containing 'ord:'-string.
ord: %askq(Give Password)%
# If the IAP does not send '>' send a CR, if or after it sends the '>'
# send a string 'pppd -f xyz.opt'.
>-""-> "pppd -f xyz.opt"
# End of script, PPP starts
```

Case 3:

A more elaborate session. The user must execute the following dialog with the IAP to start PPP negotiations:

Sender	Message
IAP	Login:
User	Myusername<CR>
IAP	Password:
User	2secret4u<CR>
IAP	#
User	ppp<CR>
IAP	Your IP address is 123.132.123.123 and please use 123.123.123.1 as your DNS server. <CR>
IAP	(begins PPP)

The following script will establish the connection:

Script sample for Case 2.

```
# Great Access service login script
# In case the line is busy
ABT "NO CARRIER" ABT BUSY
# Timeout should be set if there are problems
TOUT 25
# Send CR if Login: doesn't appear, and when it appears send
# the username
ogin:-""-ogin: Myusername
# Ask password from the user when the IAP sends a string with
# 'ord:' in it.
ord: %askq(Give Password)%
# When the IAP sends '#' send 'ppp' if it does not send '#' send a
# Carriage Return. '/' is required to distinguish the '#' character.
/#-""-/# "ppp"
# Wait for keyword " is " (/s is a space) and wait for the IP
# address (/A), then send nothing (/c)
/sis/s/A /c
# Wait for keyword " use " (/s is a space) and wait for the DNS
# address (/D), then send nothing (/c)
/suse/s/D /c
# End of script, PPP starts
```

3.6. Octal Values of Characters

If there are problems in transmitting some characters, you can always use the special sequence /ddd, where ddd is the octal value of a character.

Here are the octal values:

Non-Printing Characters

Octal	**Character**	**Remark**
000	CTRL-@	NUL (Null prompt)
001	CTRL-A	SOH (Start of heading)
002	CTRL-B	STX (Start of text)

003	CTRL-C	ETX (End of text)
004	CTRL-D	EOT (End of transmission)
005	CTRL-E	ENQ (Enquiry)
006	CTRL-F	ACK (Acknowledge)
007	CTRL-G	BEL (Bell)
008	CTRL-H	BS (Backspace)
009	CTRL-I	HT (Horizontal tab)O
010	CTRL-J	LF (Linefeed)
011	CTRL-K	VT (Vertical tab)
012	CTRL-L	NP (New page) or FF (Form feed)
013	CTRL-M	CR (Carriage return)
014	CTRL-N	SO (Shift out)
015	CTRL-O	SI (Shift in)
020	CTRL-P	DLE (Data link escape)
021	CTRL-Q	DC1 (X-ON)
022	CTRL-R	DC2
023	CTRL-S	DC3 (X-OFF)
024	CTRL-T	DC4
025	CTRL-U	NAK (No acknowledge)
026	CTRL-V	SYS (Synchronous idle)
Octal	**Character**	**Remark**
027	CTRL-W	ETB (End transmission blocks)
030	CTRL-X	CAN (Cancel)
031	CTRL-Y	EM (End of medium)
032	CTRL-Z	SUB (Substitute)
033	CTRL-[	ESC (Escape)
034	CTRL-	FS (File separator)
035	CTRL-]	GS (Group separator)
036	CTRL-^	RS (Record separator)
037	CTRL-_	US (Unit separator)
177		DEL (Delete or rubout)

Printing Characters

Octal	**Character**	**Remark**
040		Space
041	!	Exclamation Point
042	"	Double quote
043	#	Sharp Sign
044	$	Dollar Sign
045	%	Percent Sign
046	&	Ampersand
047	'	Apostrophe
050	(	Left Parenthesis

051	)	Right Parenthesis
052	*	Asterisk
053	+	Plus Sign
054	,	Comma
055	-	Hyphen
056	.	Period
057	/	Slash (Virgule)
060	0	
061	1	
062	2	
063	3	
064	4	
065	5	
066	6	
067	7	
070	8	
071	9	
072	:	Colon
073	;	Semicolon
074	<	Left Angle Bracket
075	=	Equal Sign
076	>	Right Angle Bracket
077	?	Question Mark
Octal	**Character**	**Remark**
100	@	"At" Sign
101	A	
102	B	
103	C	
104	D	
105	E	
106	F	
107	G	
110	H	
111	I	
112	J	
113	K	
114	L	
115	M	
116	N	
117	O	
120	P	
121	Q	
122	R	
123	S	

124	T	
125	U	
126	V	
127	W	
130	X	
131	Y	
132	Z	
133	[	Left Square Bracket
134	\	Back Slash
135	]	Right Square Bracket
136	^	Caret
137	_	Underscore
140	'	Back Quote
141	a	
142	b	
143	c	
144	d	
145	e	
146	f	
147	g	
150	h	
151	i	
152	j	
153	k	
154	l	
155	m	
Octal	**Character**	**Remark**
156	n	
157	o	
160	p	
161	q	
162	r	
163	s	
164	t	
165	u	
166	v	
167	w	
170	x	
171	y	
172	z	
173	{	Left Curly Brace
174	|	Vertical Bar
175	}	Right Curly Brace
176	~	Tilde

4. EXAMPLE SYSTEMS

The dial-in system to the ISP or Intranet should be chosen carefully. Several manufacturers (such as Ascend, Cisco, Hewlett Packard, Shiva, Digital, Sun and Morning Star) provide hardware and software dial-in solutions which can be used with the communicator. Local computer dealers and manufacturers' web sites may provide additional information regarding dial-in solutions.

The following examples of dial-in system configurations can be used as guidelines when creating a new dial-in system or configuring an existing one.

4.1. Microsoft RAS

This chapter will explain how to configure an existing Microsoft Windows NT 4.0 RAS service to work with the Nokia 9110 Communicator.

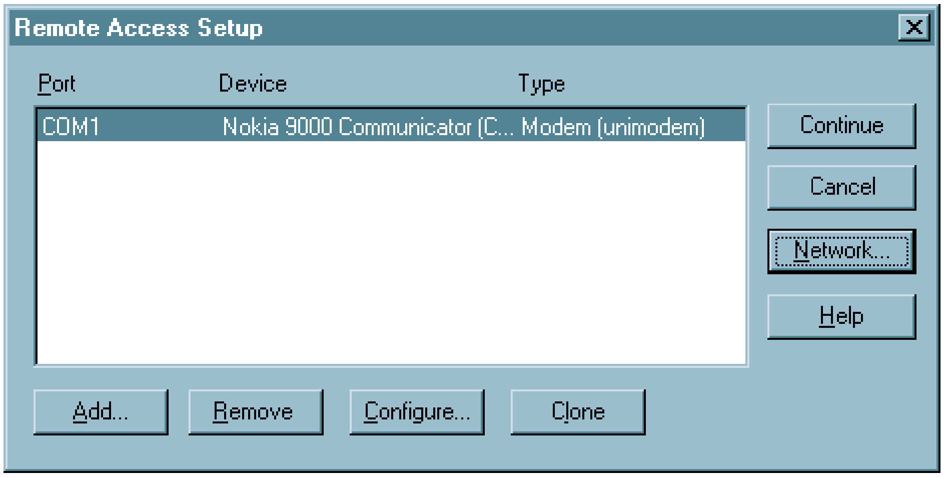
Required configurations:

- The TCP/IP settings must be configured properly on the RAS PC. These settings can vary so much that no example can be given.

- RAS must be set to "Allow any authentication including clear text", in the encryption settings.

Note: If the "Allow any authentication including clear text" is not acceptable then the RAS can be configured to use CHAP. This requires NT version 4.0 with Service Pack 2 or higher. The Service Pack information file (from http://www.microsoft.com) gives information how to set this up.

The configuration for part 2, can be found in the Control Panel's Network settings. Choose Remote Access Service (and its properties) to get to the Remote Access Setup window (Figure 4.1).



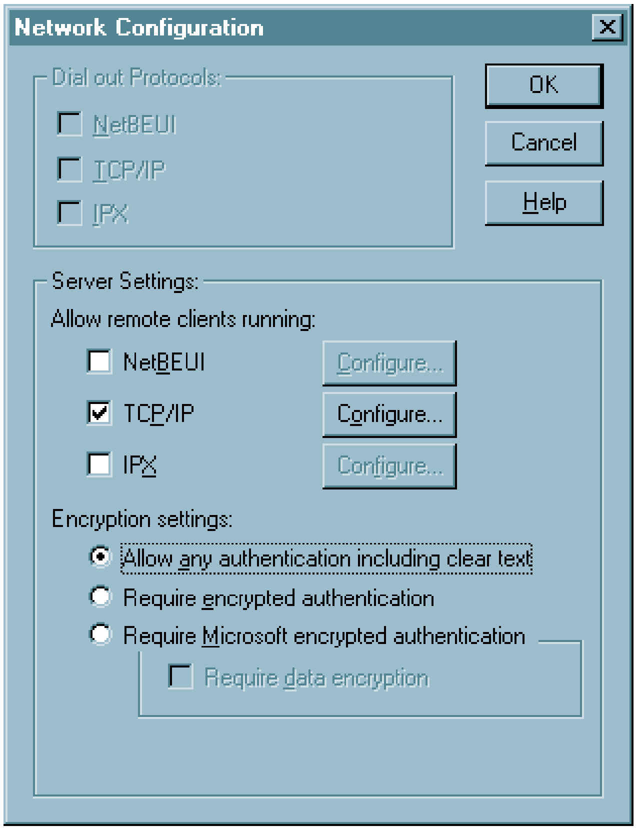
Remote Access Setup

Port	Device	Type
COM1	Nokia 9000 Communicator (C...	Modem (unimodem)

Figure 4.1 Remote Access Setup Window

Note: In this example the communicator is set up to be used as the dial-in modem via the cable.

From the Remote Access Setup window, by choosing **Network,** one can find the Encryption settings in the Network Configuration window. The communicator-supported settings for the encryption are illustrated in the Network Configuration window (Figure 4.2). The Encryption settings are configured so that RAS does not require encrypted password authentication. The dial-in client should support password encryption if encrypted passwords are required.



Network Configuration

Dial out Protocols:

□ NetBEUI

□ TCP/IP

□ IPX

Server Settings:

Allow remote clients running:

□ NetBEUI

☑ TCP/IP

□ IPX

Encryption settings:

● Allow any authentication including clear text

○ Require encrypted authentication

○ Require Microsoft encrypted authentication

□ Require data encryption

Figure 4.2 Network Configuration Window

Usually these settings are adequate to get RAS working with the communicator. In some rare cases it is necessary to remove all the settings which RAS makes to the dial-in modem when an incoming call is received. These settings can be found in the *modem.ini* file. These settings are so case-dependent that no examples can be given.

4.2. Shiva Terminal Server Settings

The Shiva server uses SPAP as an authentication protocol by default, but the communicator does not support this protocol. With Shiva, a TACACS server should be used (Nokia recommends the extended TACACS).

The following configuration is based on the Shiva Net Manager settings, with software version 4.0.1. When using conventional modems, the PPP Restart Timer must be set to 10. Automatic disconnect time should be set to 15 minutes, especially if using international data calls. That is the only major configuration needed to make the Shiva work. If using ISDN modems, then several other configuration changes are needed to make the system work. In the [Serial *] category:

- ISDNIncomingCallDefault = config
- Rate adoption: V110
- Rate: 9600

- Switch type: Euro-ISDN
- Mode: Asynchronous

With these settings in the Shiva terminal server end, and AT+CBST=71 in the communicator’s modem initialisation, the ISDN connection with Shiva works.

5. KNOWN PROBLEMS

This section presents the settings needed to solve some known connectivity problems. In some cases, it is necessary to explain the situation further so that the problem case can be solved.

5.1. ISDN Connections

ISDN dial-in lines can create interesting problems. In most cases, when the dial-in system has been configured, the only thing left to do is to set the communicator as a null modem. Thus the ISDN modem does not care about the speed of the communicator. To set up the communicator as a null modem:

> In the Internet application, choose **Settings** → choose *Internet access* → highlight the Internet access point and press **Edit** → choose *Advanced settings* → and type “AT+CBST=71” into the *Modem initialisation* field.

Note: The GSM network operator must support ISDN data calls.

5.2. Shiva Terminal Servers

The communicator supports PAP and CHAP protocols in PPP connection authentication. Shiva has its own authentication protocol called SPAP that the communicator does not support. When using a Shiva terminal server with the communicator, some configuration settings must be changed. Since SPAP cannot be used, a TACACS server should be set-up and configured to the Shiva (extended TACACS is recommended). In the communicator, the compression option must be off. The configurations are explained in more detail in section 3.

5.3. PPP Data Calls Over Long Distances

If some of the dial-up connections to the PPP service are made over exceptionally long delay links, if the PPP server uses the default 3-second restart timer, the service provider’s PPP protocol state machine may be unable to converge the option negotiations.

An example of a long delay link is a GSM data call from Hong Kong to Helsinki, Finland. The Radio Link Protocol (RLP) of GSM that is used to provide reliable asynchronous data service over the wireless link adds more delay to the satellite link delays. This problem can be avoided by configuring the PPP server to use a greater restart timer value (e.g. 11 seconds) or to configure the PPP server to use some dynamic scheme for updating the restart timer value if the server is capable of doing so.